Introduction

POur une nation, les soins de santé sont l'un des services extrêmement importants pour maintenir sa population en bonne santé et travailler pour l'essor de l'économie. Le manque de soins de santé ou l'effondrement des soins de santé peut avoir des conséquences massives sur le développement d'un pays. C'est la raison pour laquelle chaque gouvernement de chaque pays dépense des milliards de dollars pour améliorer et maintenir le fonctionnement des soins de santé afin de donner à la population des options de soins de santé abordables.

Cependant, les défis auxquels est confronté un pays comme l'Inde sont bien plus complexes que ceux du reste du monde. Dans un pays de 1,33 milliard d'habitants vivant dans des régions reculées, il est très difficile de fournir à chacun des possibilités égales et abordables en matière de soins de santé. De nombreux pauvres doivent parcourir des milliers de kilomètres pour obtenir les soins de santé dont ils ont besoin. Le manque de lits et de personnel de santé est un autre problème auquel sont confrontés les soins de santé en raison de l'augmentation de la population. Ce problème a été particulièrement observé pendant la période difficile de COVID-19, où les personnes présentant des symptômes, même légers, affluaient en grand nombre dans les établissements de santé du gouvernement, ce qui rendait la prise en charge des patients extrêmement difficile pour les fonctionnaires.

Des méthodes innovantes doivent être recherchées afin que les fonctionnaires puissent prévoir le trafic des patients. En évaluant les problèmes de santé, nous pouvons mettre au point un algorithme de ML qui peut prédire combien de temps un patient devra être hospitalisé. Si l'algorithme peut aider les fonctionnaires à obtenir une valeur approximative de la durée d'hospitalisation, ils pourront mieux gérer le trafic de patients.

1. Importation de données et librairies nescessaires

In [30]:
```
import numpy as np
import pandas as pd
#######
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
import plotly
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
plotly.offline.init_notebook_mode (connected = True)
```

In [9]:
```
#charger et lire les données
df=pd.read_csv('../data/Ifrisse_td1.csv')
#inspecter les 1ère lignes des données
df.head()
```

Out[9]:

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital
0	1	8	c	3	Z	3
1	2	2	c	5	Z	2

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital
2	3	10	e	1	X	2
3	4	26	b	2	Y	2
4	5	26	b	2	Y	2

- tableau de données : ligne = une modalité (observation) / colonne = une variable

Comme on peut le voir, nous avons de nombreuses colonnes contenant des informations telles que le code de l'hôpital, le code du type d'hôpital et de nombreux autres identifiants de ce type. Bien que nous ne disposions pas d'une documentation détaillée sur la signification de ces codes, nous les conserverons car ils pourraient nous fournir de précieuses informations pour notre apprentissage automatique.

In [51]:

```
# consulter les informations sur les données
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 318438 entries, 0 to 318437
Data columns (total 21 columns):
 #   Column                             Non-Null Count   Dtype  
---  ------                             --------------   -----  
 0   case_id                            318438 non-null  int64  
 1   Hospital_code                      318438 non-null  int64  
 2   Hospital_type_code                 318438 non-null  object 
 3   City_Code_Hospital                 318438 non-null  int64  
 4   Hospital_region_code               318438 non-null  object 
 5   Available Extra Rooms in Hospital  318438 non-null  int64  
 6   Department                         318438 non-null  object 
 7   Ward_Type                          318438 non-null  object 
 8   Ward_Facility_Code                 318438 non-null  object 
 9   Bed Grade                          318325 non-null  float64
 10  patientid                          318438 non-null  int64  
 11  City_Code_Patient                  313906 non-null  float64
 12  Type of Admission                  318438 non-null  object 
 13  Severity of Illness                318438 non-null  object 
 14  Visitors with Patient              318438 non-null  int64  
 15  Age                                318438 non-null  object 
 16  Admission_Deposit                  318438 non-null  float64
 17  Stay                               318438 non-null  object 
 18  Count                              318438 non-null  int64  
 19  Hospital_type_code_cat             318438 non-null  int64  
 20  MissingVal                         0 non-null       float64
dtypes: float64(4), int64(8), object(9)
memory usage: 51.0+ MB
```

In [49]:

```
# Analyse descriptive sur les données types "objet"
df.describe(include=['object'])
```

Out[49]:

	Hospital_type_code	Hospital_region_code	Department	Ward_Type	Ward_Facility_Code	Ty Admi
count	318438	318438	318438	318438	318438	3
unique	7	3	5	6	6	
top	a	X	gynecology	R	F	Tı
freq	143425	133336	249486	127947	112753	1!

In [50]:
```
# Analyse descriptive sur les données types "numeric"

df.describe(include=['number'])
```

Out[50]:

	case_id	Hospital_code	City_Code_Hospital	Available Extra Rooms in Hospital	Bed Grade	patient
count	318438.000000	318438.000000	318438.000000	318438.000000	318325.000000	318438.0000(
mean	159219.500000	18.318841	4.771717	3.197627	2.625807	65747.5794
std	91925.276847	8.633755	3.102535	1.168171	0.873146	37979.9364
min	1.000000	1.000000	1.000000	0.000000	1.000000	1.0000(
25%	79610.250000	11.000000	2.000000	2.000000	2.000000	32847.0000(
50%	159219.500000	19.000000	5.000000	3.000000	3.000000	65724.5000(
75%	238828.750000	26.000000	7.000000	4.000000	3.000000	98470.0000(
max	318438.000000	32.000000	13.000000	24.000000	4.000000	131624.0000(

In [ ]:
```

```

In [ ]:
```

```

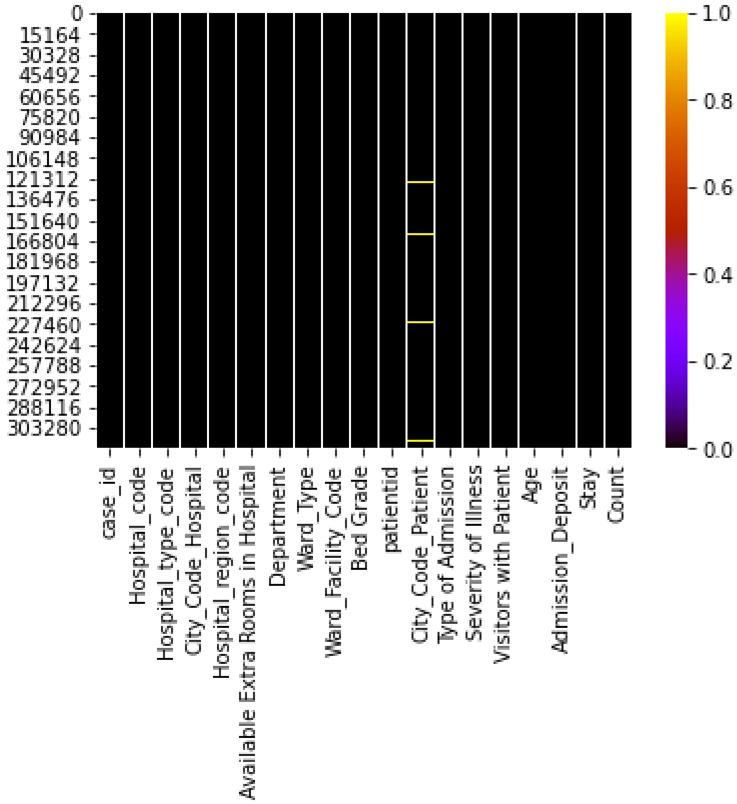
In [32]:
```
# verifier les dimensions des données
df.shape
```

Out[32]: (318438, 20)

Heatmap for missing values

In [11]:
```
# verifier les valeurs manquantes
sns.heatmap(df.isna(),cmap='gnuplot')
```

Out[11]: <AxesSubplot:>



In [44]:

```
df.isna().any()
```

Out[44]:

```
case_id                              False
Hospital_code                        False
Hospital_type_code                   False
City_Code_Hospital                   False
Hospital_region_code                 False
Available Extra Rooms in Hospital    False
Department                           False
Ward_Type                            False
Ward_Facility_Code                   False
Bed Grade                             True
patientid                            False
City_Code_Patient                     True
Type of Admission                    False
Severity of Illness                  False
Visitors with Patient                False
Age                                  False
Admission_Deposit                    False
Stay                                 False
Count                                False
Hospital_type_code_cat               False
MissingVal                            True
dtype: bool
```

In [45]:

```
# identier les types des variables de chaque colonne
df_type = pd.DataFrame(df.dtypes)
df_type
```

Out[45]:

	0
case_id	int64
Hospital_code	int64
Hospital_type_code	object

	0
City_Code_Hospital	int64
Hospital_region_code	object
Available Extra Rooms in Hospital	int64
Department	object
Ward_Type	object
Ward_Facility_Code	object
Bed Grade	float64
patientid	int64
City_Code_Patient	float64
Type of Admission	object
Severity of Illness	object
Visitors with Patient	int64
Age	object
Admission_Deposit	float64
Stay	object
Count	int64
Hospital_type_code_cat	int64
MissingVal	float64

In [46]:

```
# verifier autrement les données manquantes
df_type['MissingVal'] = df.isnull().sum()
df_type['MissingVal']
```

Out[46]:

```
case_id                                   0
Hospital_code                             0
Hospital_type_code                        0
City_Code_Hospital                        0
Hospital_region_code                      0
Available Extra Rooms in Hospital         0
Department                                0
Ward_Type                                 0
Ward_Facility_Code                        0
Bed Grade                               113
patientid                                 0
City_Code_Patient                      4532
Type of Admission                         0
Severity of Illness                       0
Visitors with Patient                     0
Age                                       0
Admission_Deposit                         0
Stay                                      0
Count                                     0
Hospital_type_code_cat                    0
MissingVal                           318438
Name: MissingVal, dtype: int64
```

In [48]:

```
# rename the 0 column
df_type = df_type.rename(columns={0:'DataT'})
```

```
df_type
```

Out[48]:

	Data	MissingVal
case_id	int64	0
Hospital_code	int64	0
Hospital_type_code	object	0
City_Code_Hospital	int64	0
Hospital_region_code	object	0
Available Extra Rooms in Hospital	int64	0
Department	object	0
Ward_Type	object	0
Ward_Facility_Code	object	0
Bed Grade	float64	113
patientid	int64	0
City_Code_Patient	float64	4532
Type of Admission	object	0
Severity of Illness	object	0
Visitors with Patient	int64	0
Age	object	0
Admission_Deposit	float64	0
Stay	object	0
Count	int64	0
Hospital_type_code_cat	int64	0
MissingVal	float64	318438

In [ ]:
```
#afficher les N 1ere lignes
```

Comme on peut le voir, la plupart des colonnes ne comportent pas de valeurs manquantes associées. C'est un très bon signe que nous traitons des données prétraitées.

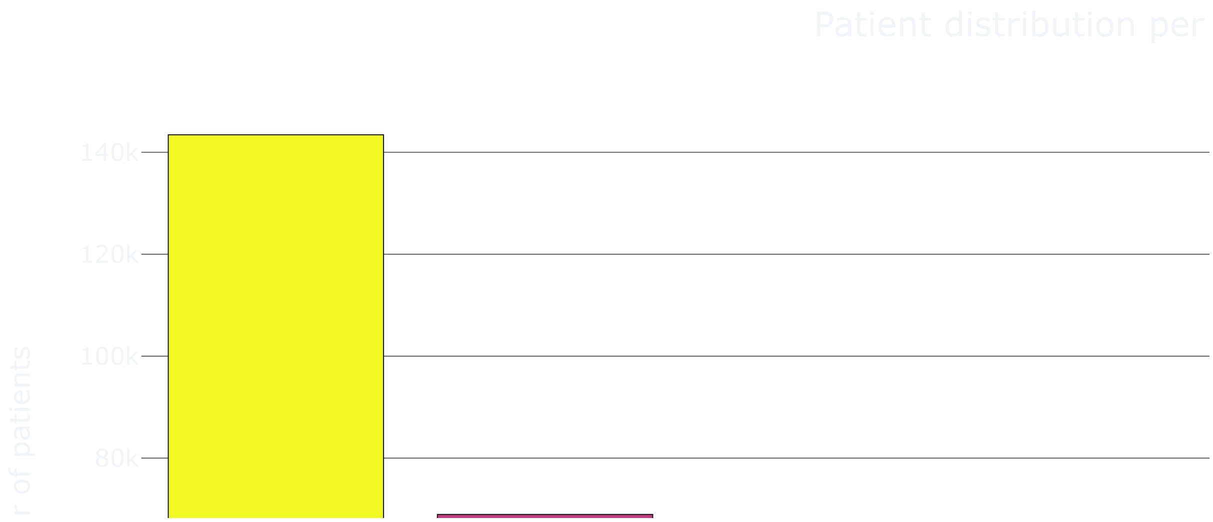
2. Data Visualisations

Hospital type code

Vérifions quel est le code des hôpitaux ayant la plus forte densité de patients.

In [13]:
```
df_hos_code=df.groupby('Hospital_type_code')['Count'].sum().reset_index().sort_v
```

In [14]:
```
fig1=px.bar(df_hos_code,x='Hospital_type_code',y='Count',color='Count',labels={'
fig1.update_layout(title='Patient distribution per hospital type code',title_x=0
fig1.show()
```

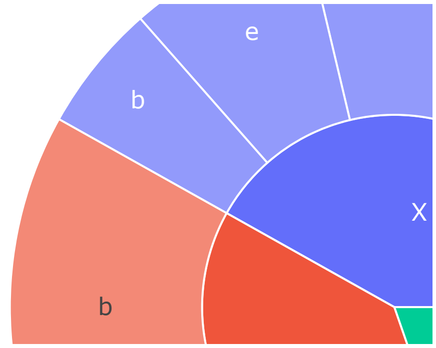


Comme on peut le voir, l'hôpital de type **A** a une charge de travail beaucoup plus élevée par rapport aux autres codes d'hôpitaux. L'hôpital de type **G** a la charge de travail la plus faible. Par conséquent, l'hôpital de type A a une probabilité beaucoup plus élevée d'être à court de lits de patients alors que le type G en a le moins. La distribution idéale aurait été une distribution uniforme où les lits sous-utilisés des autres codes de type d'hôpital auraient pu être utilisés de la même manière.

Hospital region code

Le code régional diviserait probablement tous les hôpitaux disponibles en trois régions codées X, Y et Z. Cela nous permettrait de savoir à nouveau quel type d'hôpital appartient à quelle région et les charges de travail correspondantes.

In [15]:
```
fig2=px.sunburst(df,path=['Hospital_region_code','Hospital_type_code'])
fig2.update_layout(title='Hospital region case load distribution',title_x=0.5)
fig2.show()
```



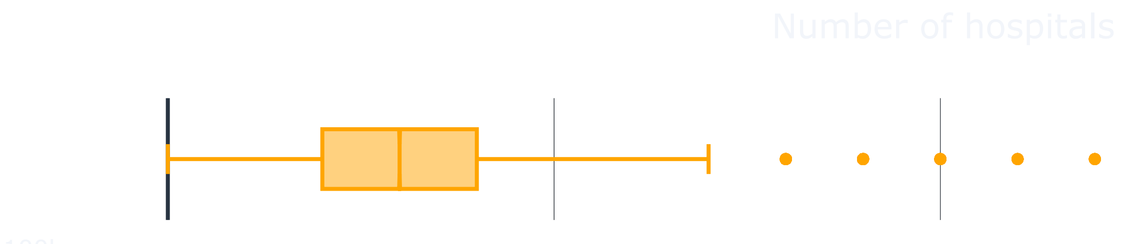
Comme on peut le voir sur le graphique ci-dessus, la charge de travail dans la région hospitalière **X** est légèrement supérieure à celle de la région hospitalière **Y** et la plus faible dans la région **Z** . Pour la région X, le type d'hôpital A a la charge de travail la plus élevée, tandis qu'elle est plus équilibrée entre a et b dans la région Y. Pour la région Z, les charges de travail les plus élevées ont été enregistrées dans le type d'hôpital C. Les hôpitaux de la région X semblent mieux répartis entre tous les types d'hôpitaux.

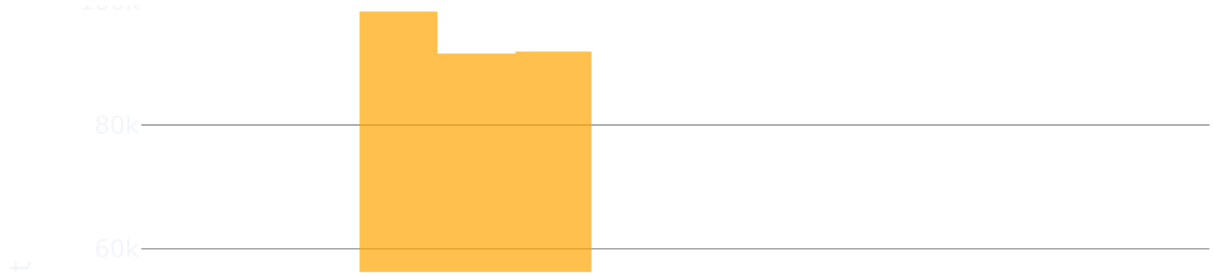
Pour la région Y, la répartition entre A et B est idéale, mais f, d et g ont été sous-utilisés. Pour la région Z, les cas de l'hôpital c étaient facilement les plus élevés, mais a et d sont extrêmement sous-utilisés.

Chambres supplémentaires disponibles à l'hôpital

C'est un indicateur important de la manière dont l'hôpital gère la charge de travail. Voyons comment les chambres supplémentaires sont réparties entre les différents hôpitaux.

In [16]:
```
fig3=px.histogram(df,x='Available Extra Rooms in Hospital',marginal='box',color_
fig3.update_layout(template='plotly_dark',title='Number of hospitals with extra
fig3.show()
```





Comme nous pouvons le voir sur l'histogramme ci-dessus, la majorité des hôpitaux disposent de chambres supplémentaires de 2, 3 ou 4 pièces. La médiane des chambres supplémentaires est de 3.

Chambres disponibles Vs Code de la région

Vérifions quelle région de X,Y et Z dispose du plus grand nombre de chambres supplémentaires. Idéalement, elles devraient avoir une répartition uniforme montrant des charges de cas égales. Vérifions ce que disent les données.

In [17]:
```
df_beds=df[['Hospital_code','Available Extra Rooms in Hospital','Hospital_regior
df_beds['Hospital_region_code']=df_beds['Hospital_region_code'].map({'X':1,'Y':2
df_beds_grouped=df_beds.groupby('Hospital_code')[['Available Extra Rooms in Hosp
```

```
/home/rodrique/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: Se
ttingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy
```

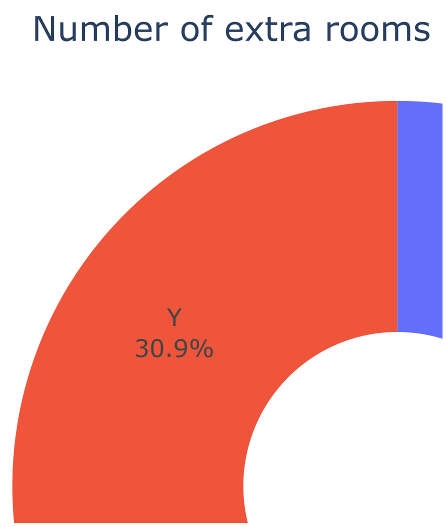
In [18]:
```
df_beds_grouped['Hospital_region_code']=df_beds_grouped['Hospital_region_code'].
df_beds_grouped.head()
```

Out[18]:

	Hospital_code	Available Extra Rooms in Hospital	Hospital_region_code
0	1	3	Y
1	2	2	Z

	Hospital_code	Available Extra Rooms in Hospital	Hospital_region_code
2	3	3	Z
3	4	3	X
4	5	3	X

In [19]:
```
df_beds_1=df_beds_grouped.groupby('Hospital_region_code')['Available Extra Rooms
fig4=px.pie(df_beds_1,values='Available Extra Rooms in Hospital',names='Hospital
fig4.update_layout(title='Number of extra rooms in each region code',title_x=0.5
fig4.update_traces(textinfo='percent+label')
```



Le graphique ci-dessus montre que le nombre de chambres supplémentaires disponibles est réparti presque égale entre les trois codes régionaux. C'est un scénario idéal.

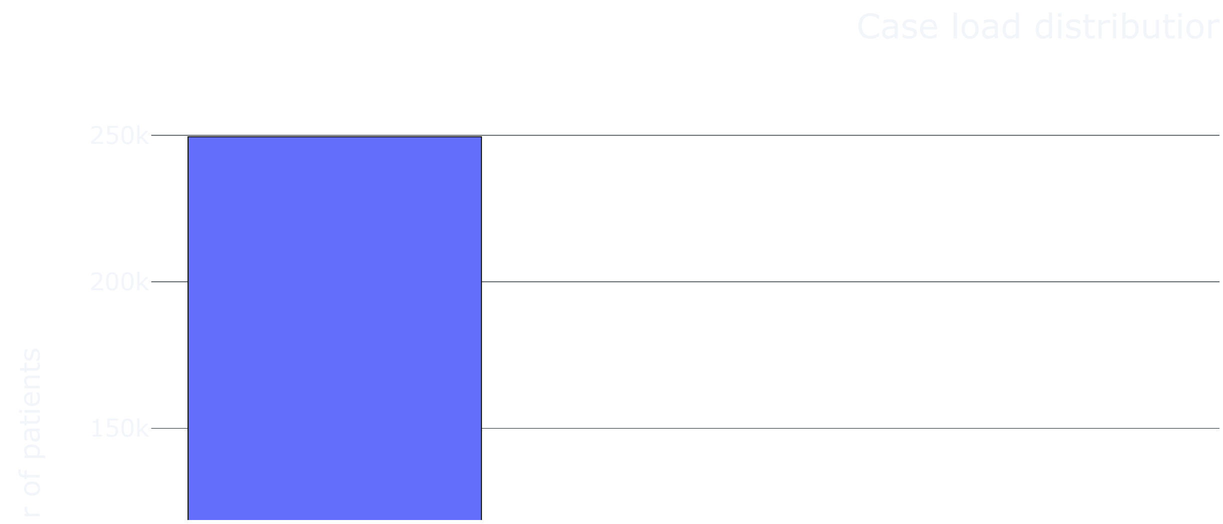
Département

Ici, nous allons vérifier quels sont les services qui ont le plus grand nombre de dossiers.

In [52]:
```
df_dept=df.groupby('Department')['Count'].sum().reset_index().sort_values(by='Co
```

In [21]:
```
fig5=px.bar(df_dept,x='Department',y='Count',color='Department',labels={'Count':
fig5.update_layout(title='Case load distribution per department',title_x=0.5,tem
```

```
fig5.show()
```



Sur le graphique ci-dessus, on voit que la majorité des patients s'inscrivent eux-mêmes au service de gynécologie. Le service de chirurgie est celui qui compte le moins de cas. On peut s'y attendre, car le nombre de naissances quotidiennes est bien plus élevé que le nombre d'opérations quotidiennes.

Gravité de la maladie

Vérifions le nombre de patients pour chaque gravité de maladie.

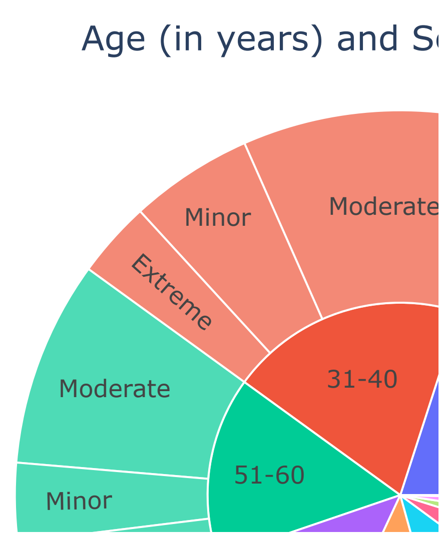
In [22]:
```
df_sev=df.groupby('Severity of Illness')['Count'].sum().reset_index().sort_value
fig6=px.funnel(df_sev,x='Count',y='Severity of Illness',labels={'Count':'Number 
fig6.update_layout(title='Case load depending upon severity of illness',title_x=
fig6.show()
```

L'entonnoir ci-dessus nous montre que la plupart des cas sont de nature modérée, puis mineurs et enfin extrêmes.

Âge contre gravité de la maladie

Vérifions s'il existe un lien entre l'âge et la gravité de la maladie

In [23]:
```
fig7=px.sunburst(df,path=['Age','Severity of Illness'])
fig7.update_layout(title='Age (in years) and Severity of Illness',title_x=0.5)
fig7.show()
```

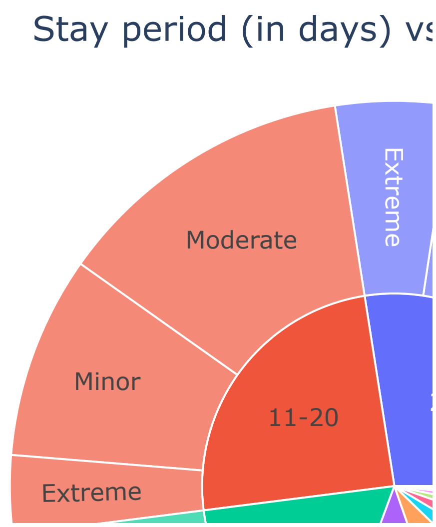


Comme nous pouvons le voir sur le graphique des rayons du soleil ci-dessus, les cas extrêmes sont moins nombreux pour chaque groupe d'âge. Cependant, les cas extrêmes par rapport aux cas modérés et mineurs semblent augmenter avec les groupes d'âge plus élevés.

Gravité de la maladie contre séjour

Voyons comment la gravité de la maladie et la durée du séjour évoluent.

In [24]:
```
fig8=px.sunburst(df,path=['Stay','Severity of Illness'])
fig8.update_layout(title='Stay period (in days) vs Severity of illness',title_x=
fig8.show()
```



Le graphique ci-dessus montre qu'en général, pour les séjours de courte durée (jusqu'à 20 jours), la gravité de la maladie est mineure ou modérée. Pour des séjours plus longs, les cas d'extrême gravité commencent à augmenter.

Les durées de séjour les plus courantes dans les hôpitaux sont de 21 à 30 jours et de 11 à 20 jours.

3. Discussion sur la qualité des données

Comme on peut le voir, nous disposons de nombreuses données. Certaines de ces données nous fournissent des informations utiles tandis que d'autres ne le font pas. Une partie des données n'est pas numérique et, à cette fin, nous devons travailler avec les données pour les coder en les étiquettant autrement (transformation de variables).

In [18]:

```
df.head()
```

Out[18]:

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available Extra Rooms in Hospital
0	1	8	c	3	Z	3
1	2	2	c	5	Z	2
2	3	10	e	1	X	2
3	4	26	b	2	Y	2
4	5	26	b	2	Y	2

In [26]:

```
from sklearn.preprocessing import LabelEncoder
```

In [29]:

```
le=LabelEncoder()
df['Hospital_type_code_cat']=le.fit_transform(df['Hospital_type_code'])
print(df['Hospital_type_code_cat'])
df['Hospital_type_code'] #nominal to discrete
```

```
0         2
1         2
2         4
3         1
4         1
         ..
318433    0
318434    0
318435    0
318436    1
318437    0
Name: Hospital_type_code_cat, Length: 318438, dtype: int64
```

Out[29]:

```
0         c
1         c
2         e
```

```
3         b
4         b
         ..
318433    a
318434    a
318435    a
318436    b
318437    a
Name: Hospital_type_code, Length: 318438, dtype: object
```

In [21]:
```
df_reg_codes=pd.get_dummies(df['Hospital_region_code'])

df=pd.merge(df,df_reg_codes,on=df.index)
```

In [22]:
```
df.drop('key_0',axis=1,inplace=True)
```

In [23]:
```
df_dept=pd.get_dummies(df['Department'])
df=pd.merge(df,df_dept,on=df.index)
df.drop('key_0',axis=1,inplace=True)
```

In [24]:
```
df['Ward_Type_cat']=le.fit_transform(df['Ward_Type'])
df['Ward_Facility_Code_cat']=le.fit_transform(df['Ward_Facility_Code'])
```

In [25]:
```
df_adm=pd.get_dummies(df['Type of Admission'])
df=pd.merge(df,df_adm,on=df.index)
df.drop('key_0',axis=1,inplace=True)
df_sev_ill=pd.get_dummies(df['Severity of Illness'])
df=pd.merge(df,df_sev_ill,on=df.index)
df.drop('key_0',axis=1,inplace=True)
```

In [26]:
```
df['Age_cat']=le.fit_transform(df['Age'])
df['Stay_cat']=le.fit_transform(df['Stay'])
```

Maintenant que nous avons soit codé l'étiquette, soit codé en une seule fois toutes les données, supprimons les colonnes catégorielles précédentes de la trame de données.

In [27]:
```
df_train=df.copy()
```

In [28]:
```
unn_cols=['case_id','Hospital_type_code','Hospital_region_code','Department','Wa
          'Ward_Facility_Code','Type of Admission','Severity of Illness','Age','
          'Count','Admission_Deposit','Bed Grade','City_Code_Patient']

for cols in unn_cols:
    df_train.drop(cols,axis=1,inplace=True)
```

In [29]:
```
df_train.info()
```

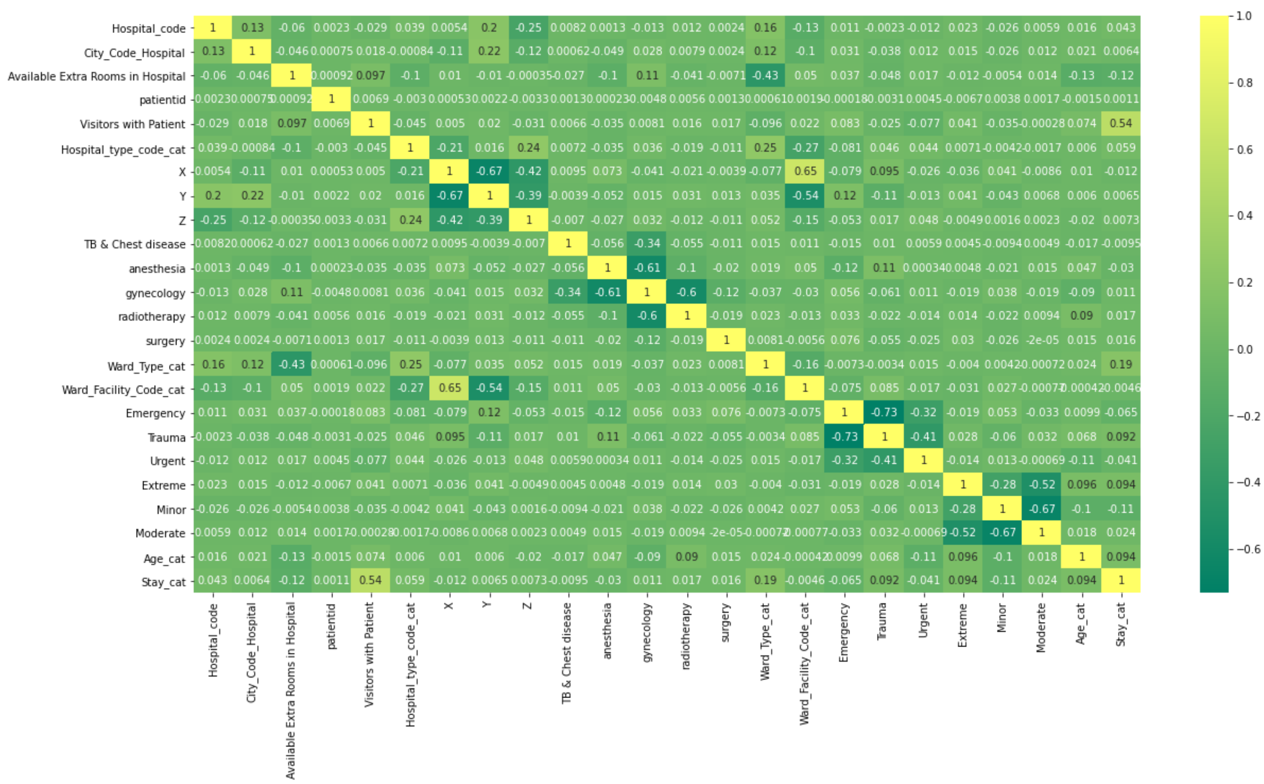
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 318438 entries, 0 to 318437
Data columns (total 24 columns):
 #   Column                             Non-Null Count   Dtype
---  ------                             --------------   -----
 0   Hospital_code                      318438 non-null  int64
 1   City_Code_Hospital                 318438 non-null  int64
 2   Available Extra Rooms in Hospital  318438 non-null  int64
 3   patientid                          318438 non-null  int64
 4   Visitors with Patient              318438 non-null  int64
 5   Hospital_type_code_cat             318438 non-null  int64
 6   X                                  318438 non-null  uint8
```

```
 7   Y                                  318438 non-null  uint8
 8   Z                                  318438 non-null  uint8
 9   TB & Chest disease                 318438 non-null  uint8
 10  anesthesia                         318438 non-null  uint8
 11  gynecology                         318438 non-null  uint8
 12  radiotherapy                       318438 non-null  uint8
 13  surgery                            318438 non-null  uint8
 14  Ward_Type_cat                      318438 non-null  int64
 15  Ward_Facility_Code_cat             318438 non-null  int64
 16  Emergency                          318438 non-null  uint8
 17  Trauma                             318438 non-null  uint8
 18  Urgent                             318438 non-null  uint8
 19  Extreme                            318438 non-null  uint8
 20  Minor                              318438 non-null  uint8
 21  Moderate                           318438 non-null  uint8
 22  Age_cat                            318438 non-null  int64
 23  Stay_cat                           318438 non-null  int64
dtypes: int64(10), uint8(14)
memory usage: 31.0 MB
```

Comme nous pouvons le voir, toutes les données disponibles sont maintenant sous forme numérique et peuvent donc être utilisées pour les algorithmes ML. Vérifions la corrélation de chaque élément entre eux grâce à une carte thermique.

In [30]:
```python
correlations=df_train.corr()
plt.figure(figsize=(20,10))
sns.heatmap(correlations,cmap='summer',annot=True,fmt='.2g')
```

Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7f45840f7f90>



4. Machine Learning

Dans cette section, nous allons appliquer certains des algorithmes ML pour une classification correcte.

Training phase

a) Logistic Regression

In [31]:
```
target=df_train['Stay_cat']
train_df=df_train.iloc[:,:-1]
```

In [32]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
reg_log=LogisticRegression()
```

In [33]:
```
X_train,X_test,y_train,y_test=train_test_split(train_df,target,test_size=0.3,shu
```

In [34]:
```
reg_log.fit(X_train,y_train)
```

Out[34]:
```
LogisticRegression()
```

In [35]:
```
reg_log.score(X_train,y_train)
```

Out[35]:
```
0.27111428135626675
```

In [36]:
```
y_predict_log=reg_log.predict(X_test)
reg_log.score(X_test,y_test)
```

Out[36]:
```
0.26826613072059624
```

As we can see, the logistic regression has performed extremely poorly with only 28 % accuracy. Let us check another algorithm.

b) Random Forest Classifier

In [37]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [38]:
```
rfc=RandomForestClassifier(max_depth=10,random_state=0)
```

In [39]:
```
rfc.fit(X_train,y_train)
rfc.score(X_train,y_train)
```

Out[39]:
```
0.4094147308731035
```

In [40]:
```
y_preds_rfc=rfc.predict(X_test)
rfc.score(X_test,y_test)
```

Out[40]:
```
0.39467403592513506
```

We can see that the scores for Random Forest Classifier are better than logistic regression for a pre pruned max_depth of 10.

c) CatBoost

In [41]:
```python
from catboost import CatBoostClassifier
```

In [42]:
```python
cb_clf=CatBoostClassifier(iterations=800,
                          learning_rate=0.08,
                          depth=8,
                          loss_function='MultiClass',
                          eval_metric='Accuracy')
```

In [43]:
```python
cb_clf.fit(X_train,y_train)
cb_clf.score(X_train,y_train)
```

```
0:	learn: 0.3646963	total: 758ms	remaining: 10m 5s
1:	learn: 0.3760913	total: 1.27s	remaining: 8m 28s
2:	learn: 0.3762124	total: 1.79s	remaining: 7m 56s
3:	learn: 0.3789490	total: 2.3s	remaining: 7m 38s
4:	learn: 0.3804249	total: 2.82s	remaining: 7m 29s
5:	learn: 0.3816138	total: 3.34s	remaining: 7m 22s
6:	learn: 0.3816497	total: 3.85s	remaining: 7m 16s
7:	learn: 0.3828520	total: 4.36s	remaining: 7m 11s
8:	learn: 0.3829955	total: 4.88s	remaining: 7m 9s
9:	learn: 0.3828834	total: 5.46s	remaining: 7m 11s
10:	learn: 0.3835428	total: 5.98s	remaining: 7m 9s
11:	learn: 0.3841933	total: 6.48s	remaining: 7m 5s
12:	learn: 0.3844894	total: 6.99s	remaining: 7m 3s
13:	learn: 0.3851040	total: 7.5s	remaining: 7m
14:	learn: 0.3862256	total: 8s	remaining: 6m 58s
15:	learn: 0.3873606	total: 8.51s	remaining: 6m 56s
16:	learn: 0.3882803	total: 9.01s	remaining: 6m 55s
17:	learn: 0.3887154	total: 9.53s	remaining: 6m 54s
18:	learn: 0.3897831	total: 10s	remaining: 6m 52s
19:	learn: 0.3899985	total: 10.5s	remaining: 6m 51s
20:	learn: 0.3901824	total: 11s	remaining: 6m 49s
21:	learn: 0.3907880	total: 11.6s	remaining: 6m 48s
22:	learn: 0.3909989	total: 12.1s	remaining: 6m 47s
23:	learn: 0.3915238	total: 12.6s	remaining: 6m 46s
24:	learn: 0.3917930	total: 13.1s	remaining: 6m 45s
25:	learn: 0.3917930	total: 13.6s	remaining: 6m 44s
26:	learn: 0.3927575	total: 14.1s	remaining: 6m 43s
27:	learn: 0.3932196	total: 14.6s	remaining: 6m 42s
28:	learn: 0.3935695	total: 15.1s	remaining: 6m 41s
29:	learn: 0.3940181	total: 15.6s	remaining: 6m 40s
30:	learn: 0.3946551	total: 16.1s	remaining: 6m 40s
31:	learn: 0.3947045	total: 16.7s	remaining: 6m 41s
32:	learn: 0.3949198	total: 17.3s	remaining: 6m 42s
33:	learn: 0.3948391	total: 18.2s	remaining: 6m 49s
34:	learn: 0.3953595	total: 18.8s	remaining: 6m 51s
35:	learn: 0.3955569	total: 19.4s	remaining: 6m 50s
36:	learn: 0.3958350	total: 20s	remaining: 6m 51s
37:	learn: 0.3958260	total: 20.5s	remaining: 6m 51s
38:	learn: 0.3961266	total: 21s	remaining: 6m 50s
39:	learn: 0.3962836	total: 21.6s	remaining: 6m 49s
40:	learn: 0.3965618	total: 22.1s	remaining: 6m 48s
41:	learn: 0.3971943	total: 22.6s	remaining: 6m 47s
42:	learn: 0.3974186	total: 23.1s	remaining: 6m 46s
43:	learn: 0.3974411	total: 23.6s	remaining: 6m 45s
44:	learn: 0.3975532	total: 24.1s	remaining: 6m 44s
45:	learn: 0.3976923	total: 24.7s	remaining: 6m 44s
46:	learn: 0.3977955	total: 25.2s	remaining: 6m 43s
47:	learn: 0.3979256	total: 25.7s	remaining: 6m 42s
48:	learn: 0.3981634	total: 26.2s	remaining: 6m 41s
49:	learn: 0.3982351	total: 26.7s	remaining: 6m 41s
50:	learn: 0.3984056	total: 27.3s	remaining: 6m 40s
51:	learn: 0.3985357	total: 27.9s	remaining: 6m 40s
52:	learn: 0.3985537	total: 28.4s	remaining: 6m 39s
```

```
53:	learn: 0.3985761	total: 28.9s	remaining: 6m 39s
54:	learn: 0.3985895	total: 29.4s	remaining: 6m 38s
55:	learn: 0.3988183	total: 30s	remaining: 6m 38s
56:	learn: 0.3991458	total: 30.6s	remaining: 6m 38s
57:	learn: 0.3992535	total: 31.2s	remaining: 6m 39s
58:	learn: 0.3993701	total: 31.9s	remaining: 6m 40s
59:	learn: 0.3994329	total: 32.6s	remaining: 6m 41s
60:	learn: 0.3996797	total: 33.2s	remaining: 6m 41s
61:	learn: 0.3997829	total: 33.7s	remaining: 6m 40s
62:	learn: 0.3998995	total: 34.2s	remaining: 6m 39s
63:	learn: 0.3999982	total: 34.7s	remaining: 6m 39s
64:	learn: 0.4001911	total: 35.2s	remaining: 6m 38s
65:	learn: 0.4002315	total: 35.7s	remaining: 6m 37s
66:	learn: 0.4004513	total: 36.2s	remaining: 6m 36s
67:	learn: 0.4005680	total: 36.7s	remaining: 6m 35s
68:	learn: 0.4006308	total: 37.2s	remaining: 6m 34s
69:	learn: 0.4008147	total: 37.7s	remaining: 6m 33s
70:	learn: 0.4008102	total: 38.2s	remaining: 6m 32s
71:	learn: 0.4009493	total: 38.8s	remaining: 6m 32s
72:	learn: 0.4009762	total: 39.3s	remaining: 6m 31s
73:	learn: 0.4009897	total: 39.9s	remaining: 6m 31s
74:	learn: 0.4011870	total: 40.4s	remaining: 6m 30s
75:	learn: 0.4014338	total: 40.9s	remaining: 6m 29s
76:	learn: 0.4017299	total: 41.4s	remaining: 6m 28s
77:	learn: 0.4018689	total: 41.9s	remaining: 6m 27s
78:	learn: 0.4017972	total: 42.4s	remaining: 6m 26s
79:	learn: 0.4020125	total: 42.9s	remaining: 6m 26s
80:	learn: 0.4020574	total: 43.4s	remaining: 6m 25s
81:	learn: 0.4021875	total: 43.9s	remaining: 6m 24s
82:	learn: 0.4022862	total: 44.4s	remaining: 6m 23s
83:	learn: 0.4023669	total: 44.9s	remaining: 6m 22s
84:	learn: 0.4026944	total: 45.4s	remaining: 6m 22s
85:	learn: 0.4029636	total: 45.9s	remaining: 6m 21s
86:	learn: 0.4029995	total: 46.4s	remaining: 6m 20s
87:	learn: 0.4031296	total: 46.9s	remaining: 6m 19s
88:	learn: 0.4033000	total: 47.5s	remaining: 6m 19s
89:	learn: 0.4033898	total: 48s	remaining: 6m 18s
90:	learn: 0.4034481	total: 48.5s	remaining: 6m 17s
91:	learn: 0.4035558	total: 49s	remaining: 6m 16s
92:	learn: 0.4036679	total: 49.5s	remaining: 6m 16s
93:	learn: 0.4038339	total: 50.1s	remaining: 6m 16s
94:	learn: 0.4038967	total: 50.7s	remaining: 6m 15s
95:	learn: 0.4040941	total: 51.4s	remaining: 6m 16s
96:	learn: 0.4042063	total: 51.9s	remaining: 6m 16s
97:	learn: 0.4043094	total: 52.5s	remaining: 6m 15s
98:	learn: 0.4044754	total: 53s	remaining: 6m 15s
99:	learn: 0.4045831	total: 53.5s	remaining: 6m 14s
100:	learn: 0.4047132	total: 54s	remaining: 6m 13s
101:	learn: 0.4047895	total: 54.5s	remaining: 6m 13s
102:	learn: 0.4048074	total: 55s	remaining: 6m 12s
103:	learn: 0.4049958	total: 55.5s	remaining: 6m 11s
104:	learn: 0.4049644	total: 56s	remaining: 6m 10s
105:	learn: 0.4050048	total: 56.6s	remaining: 6m 10s
106:	learn: 0.4051259	total: 57.1s	remaining: 6m 9s
107:	learn: 0.4052336	total: 57.6s	remaining: 6m 8s
108:	learn: 0.4054444	total: 58.1s	remaining: 6m 8s
109:	learn: 0.4053996	total: 58.6s	remaining: 6m 7s
110:	learn: 0.4056777	total: 59.1s	remaining: 6m 6s
111:	learn: 0.4057136	total: 59.6s	remaining: 6m 6s
112:	learn: 0.4059245	total: 1m	remaining: 6m 5s
113:	learn: 0.4059469	total: 1m	remaining: 6m 5s
114:	learn: 0.4064045	total: 1m 1s	remaining: 6m 4s
115:	learn: 0.4064942	total: 1m 1s	remaining: 6m 3s
116:	learn: 0.4065929	total: 1m 2s	remaining: 6m 3s
117:	learn: 0.4066737	total: 1m 2s	remaining: 6m 2s
```

```
118:	learn: 0.4068935	total: 1m 3s	remaining: 6m 2s
119:	learn: 0.4070415	total: 1m 4s	remaining: 6m 2s
120:	learn: 0.4068127	total: 1m 4s	remaining: 6m 2s
121:	learn: 0.4070146	total: 1m 5s	remaining: 6m 3s
122:	learn: 0.4071941	total: 1m 5s	remaining: 6m 2s
123:	learn: 0.4075844	total: 1m 6s	remaining: 6m 2s
124:	learn: 0.4077234	total: 1m 6s	remaining: 6m 1s
125:	learn: 0.4078535	total: 1m 7s	remaining: 6m
126:	learn: 0.4077952	total: 1m 7s	remaining: 5m 59s
127:	learn: 0.4079298	total: 1m 8s	remaining: 5m 59s
128:	learn: 0.4080958	total: 1m 8s	remaining: 5m 58s
129:	learn: 0.4082349	total: 1m 9s	remaining: 5m 57s
130:	learn: 0.4081586	total: 1m 9s	remaining: 5m 57s
131:	learn: 0.4081451	total: 1m 10s	remaining: 5m 56s
132:	learn: 0.4082035	total: 1m 10s	remaining: 5m 55s
133:	learn: 0.4081721	total: 1m 11s	remaining: 5m 55s
134:	learn: 0.4083380	total: 1m 12s	remaining: 5m 54s
135:	learn: 0.4083964	total: 1m 12s	remaining: 5m 54s
136:	learn: 0.4084592	total: 1m 13s	remaining: 5m 53s
137:	learn: 0.4084592	total: 1m 13s	remaining: 5m 52s
138:	learn: 0.4086252	total: 1m 14s	remaining: 5m 52s
139:	learn: 0.4087104	total: 1m 14s	remaining: 5m 51s
140:	learn: 0.4087597	total: 1m 15s	remaining: 5m 50s
141:	learn: 0.4089078	total: 1m 15s	remaining: 5m 50s
142:	learn: 0.4090155	total: 1m 16s	remaining: 5m 49s
143:	learn: 0.4091142	total: 1m 16s	remaining: 5m 49s
144:	learn: 0.4092801	total: 1m 17s	remaining: 5m 48s
145:	learn: 0.4094237	total: 1m 17s	remaining: 5m 48s
146:	learn: 0.4094237	total: 1m 18s	remaining: 5m 48s
147:	learn: 0.4095045	total: 1m 18s	remaining: 5m 47s
148:	learn: 0.4095807	total: 1m 19s	remaining: 5m 47s
149:	learn: 0.4096076	total: 1m 20s	remaining: 5m 46s
150:	learn: 0.4097108	total: 1m 20s	remaining: 5m 46s
151:	learn: 0.4096974	total: 1m 21s	remaining: 5m 46s
152:	learn: 0.4097557	total: 1m 21s	remaining: 5m 45s
153:	learn: 0.4097557	total: 1m 22s	remaining: 5m 45s
154:	learn: 0.4100473	total: 1m 22s	remaining: 5m 44s
155:	learn: 0.4101505	total: 1m 23s	remaining: 5m 44s
156:	learn: 0.4103075	total: 1m 24s	remaining: 5m 44s
157:	learn: 0.4105991	total: 1m 24s	remaining: 5m 44s
158:	learn: 0.4106305	total: 1m 25s	remaining: 5m 43s
159:	learn: 0.4107696	total: 1m 25s	remaining: 5m 43s
160:	learn: 0.4107471	total: 1m 26s	remaining: 5m 42s
161:	learn: 0.4109490	total: 1m 26s	remaining: 5m 42s
162:	learn: 0.4111419	total: 1m 27s	remaining: 5m 41s
163:	learn: 0.4113259	total: 1m 27s	remaining: 5m 40s
164:	learn: 0.4114021	total: 1m 28s	remaining: 5m 40s
165:	learn: 0.4114874	total: 1m 28s	remaining: 5m 39s
166:	learn: 0.4114560	total: 1m 29s	remaining: 5m 38s
167:	learn: 0.4115995	total: 1m 29s	remaining: 5m 38s
168:	learn: 0.4117117	total: 1m 30s	remaining: 5m 37s
169:	learn: 0.4118552	total: 1m 30s	remaining: 5m 37s
170:	learn: 0.4120840	total: 1m 31s	remaining: 5m 36s
171:	learn: 0.4120840	total: 1m 31s	remaining: 5m 35s
172:	learn: 0.4122186	total: 1m 32s	remaining: 5m 35s
173:	learn: 0.4125282	total: 1m 32s	remaining: 5m 34s
174:	learn: 0.4125910	total: 1m 33s	remaining: 5m 34s
175:	learn: 0.4125640	total: 1m 34s	remaining: 5m 33s
176:	learn: 0.4127076	total: 1m 34s	remaining: 5m 32s
177:	learn: 0.4128377	total: 1m 35s	remaining: 5m 32s
178:	learn: 0.4129319	total: 1m 35s	remaining: 5m 31s
179:	learn: 0.4131697	total: 1m 36s	remaining: 5m 31s
180:	learn: 0.4132549	total: 1m 36s	remaining: 5m 31s
181:	learn: 0.4132953	total: 1m 37s	remaining: 5m 31s
182:	learn: 0.4134388	total: 1m 38s	remaining: 5m 31s
```

```
183:	learn: 0.4133446	total: 1m 38s	remaining: 5m 30s
184:	learn: 0.4134703	total: 1m 39s	remaining: 5m 30s
185:	learn: 0.4136183	total: 1m 39s	remaining: 5m 29s
186:	learn: 0.4137215	total: 1m 40s	remaining: 5m 28s
187:	learn: 0.4137484	total: 1m 40s	remaining: 5m 28s
188:	learn: 0.4137933	total: 1m 41s	remaining: 5m 27s
189:	learn: 0.4137977	total: 1m 41s	remaining: 5m 26s
190:	learn: 0.4139144	total: 1m 42s	remaining: 5m 26s
191:	learn: 0.4140714	total: 1m 42s	remaining: 5m 25s
192:	learn: 0.4142194	total: 1m 43s	remaining: 5m 25s
193:	learn: 0.4142867	total: 1m 43s	remaining: 5m 24s
194:	learn: 0.4144034	total: 1m 44s	remaining: 5m 24s
195:	learn: 0.4144931	total: 1m 44s	remaining: 5m 23s
196:	learn: 0.4145245	total: 1m 45s	remaining: 5m 22s
197:	learn: 0.4146367	total: 1m 45s	remaining: 5m 22s
198:	learn: 0.4146681	total: 1m 46s	remaining: 5m 21s
199:	learn: 0.4148520	total: 1m 46s	remaining: 5m 20s
200:	learn: 0.4149642	total: 1m 47s	remaining: 5m 20s
201:	learn: 0.4148565	total: 1m 47s	remaining: 5m 19s
202:	learn: 0.4149686	total: 1m 48s	remaining: 5m 19s
203:	learn: 0.4151122	total: 1m 49s	remaining: 5m 18s
204:	learn: 0.4151795	total: 1m 49s	remaining: 5m 17s
205:	learn: 0.4152692	total: 1m 50s	remaining: 5m 17s
206:	learn: 0.4153724	total: 1m 50s	remaining: 5m 16s
207:	learn: 0.4154666	total: 1m 51s	remaining: 5m 16s
208:	learn: 0.4155653	total: 1m 51s	remaining: 5m 15s
209:	learn: 0.4156550	total: 1m 52s	remaining: 5m 14s
210:	learn: 0.4157313	total: 1m 52s	remaining: 5m 14s
211:	learn: 0.4158210	total: 1m 53s	remaining: 5m 13s
212:	learn: 0.4159735	total: 1m 53s	remaining: 5m 12s
213:	learn: 0.4161979	total: 1m 54s	remaining: 5m 12s
214:	learn: 0.4162652	total: 1m 54s	remaining: 5m 11s
215:	learn: 0.4164401	total: 1m 55s	remaining: 5m 11s
216:	learn: 0.4163728	total: 1m 55s	remaining: 5m 10s
217:	learn: 0.4163190	total: 1m 56s	remaining: 5m 10s
218:	learn: 0.4164042	total: 1m 56s	remaining: 5m 10s
219:	learn: 0.4164446	total: 1m 57s	remaining: 5m 9s
220:	learn: 0.4166420	total: 1m 58s	remaining: 5m 9s
221:	learn: 0.4167183	total: 1m 58s	remaining: 5m 8s
222:	learn: 0.4166869	total: 1m 59s	remaining: 5m 8s
223:	learn: 0.4168484	total: 1m 59s	remaining: 5m 7s
224:	learn: 0.4169560	total: 2m	remaining: 5m 6s
225:	learn: 0.4170772	total: 2m	remaining: 5m 6s
226:	learn: 0.4171444	total: 2m 1s	remaining: 5m 5s
227:	learn: 0.4172431	total: 2m 1s	remaining: 5m 5s
228:	learn: 0.4171669	total: 2m 2s	remaining: 5m 4s
229:	learn: 0.4171983	total: 2m 2s	remaining: 5m 3s
230:	learn: 0.4173688	total: 2m 3s	remaining: 5m 3s
231:	learn: 0.4174989	total: 2m 3s	remaining: 5m 2s
232:	learn: 0.4175482	total: 2m 4s	remaining: 5m 2s
233:	learn: 0.4175392	total: 2m 4s	remaining: 5m 1s
234:	learn: 0.4176334	total: 2m 5s	remaining: 5m
235:	learn: 0.4177277	total: 2m 5s	remaining: 5m
236:	learn: 0.4178622	total: 2m 6s	remaining: 4m 59s
237:	learn: 0.4178219	total: 2m 6s	remaining: 4m 59s
238:	learn: 0.4178712	total: 2m 7s	remaining: 4m 58s
239:	learn: 0.4179340	total: 2m 7s	remaining: 4m 58s
240:	learn: 0.4180507	total: 2m 8s	remaining: 4m 57s
241:	learn: 0.4182256	total: 2m 8s	remaining: 4m 57s
242:	learn: 0.4183378	total: 2m 9s	remaining: 4m 56s
243:	learn: 0.4183647	total: 2m 10s	remaining: 4m 56s
244:	learn: 0.4185217	total: 2m 10s	remaining: 4m 56s
245:	learn: 0.4185486	total: 2m 11s	remaining: 4m 56s
246:	learn: 0.4186114	total: 2m 11s	remaining: 4m 55s
247:	learn: 0.4186339	total: 2m 12s	remaining: 4m 54s
```

```
248:	learn: 0.4188088	total: 2m 13s	remaining: 4m 54s
249:	learn: 0.4189614	total: 2m 13s	remaining: 4m 53s
250:	learn: 0.4191229	total: 2m 14s	remaining: 4m 53s
251:	learn: 0.4192216	total: 2m 14s	remaining: 4m 52s
252:	learn: 0.4191677	total: 2m 15s	remaining: 4m 51s
253:	learn: 0.4192440	total: 2m 15s	remaining: 4m 51s
254:	learn: 0.4194459	total: 2m 16s	remaining: 4m 50s
255:	learn: 0.4197823	total: 2m 16s	remaining: 4m 50s
256:	learn: 0.4199259	total: 2m 17s	remaining: 4m 49s
257:	learn: 0.4201816	total: 2m 17s	remaining: 4m 49s
258:	learn: 0.4201323	total: 2m 18s	remaining: 4m 49s
259:	learn: 0.4202399	total: 2m 18s	remaining: 4m 48s
260:	learn: 0.4202444	total: 2m 19s	remaining: 4m 48s
261:	learn: 0.4205405	total: 2m 20s	remaining: 4m 47s
262:	learn: 0.4206751	total: 2m 20s	remaining: 4m 46s
263:	learn: 0.4209308	total: 2m 21s	remaining: 4m 46s
264:	learn: 0.4209936	total: 2m 21s	remaining: 4m 46s
265:	learn: 0.4210654	total: 2m 22s	remaining: 4m 45s
266:	learn: 0.4211237	total: 2m 22s	remaining: 4m 45s
267:	learn: 0.4210968	total: 2m 23s	remaining: 4m 44s
268:	learn: 0.4214467	total: 2m 23s	remaining: 4m 44s
269:	learn: 0.4217204	total: 2m 24s	remaining: 4m 43s
270:	learn: 0.4217293	total: 2m 25s	remaining: 4m 43s
271:	learn: 0.4218101	total: 2m 25s	remaining: 4m 42s
272:	learn: 0.4218729	total: 2m 26s	remaining: 4m 41s
273:	learn: 0.4219043	total: 2m 26s	remaining: 4m 41s
274:	learn: 0.4220568	total: 2m 27s	remaining: 4m 40s
275:	learn: 0.4220389	total: 2m 27s	remaining: 4m 40s
276:	learn: 0.4221824	total: 2m 28s	remaining: 4m 39s
277:	learn: 0.4221645	total: 2m 28s	remaining: 4m 39s
278:	learn: 0.4223978	total: 2m 29s	remaining: 4m 38s
279:	learn: 0.4225010	total: 2m 29s	remaining: 4m 38s
280:	learn: 0.4225548	total: 2m 30s	remaining: 4m 38s
281:	learn: 0.4226714	total: 2m 31s	remaining: 4m 37s
282:	learn: 0.4228060	total: 2m 31s	remaining: 4m 37s
283:	learn: 0.4230258	total: 2m 32s	remaining: 4m 36s
284:	learn: 0.4230931	total: 2m 32s	remaining: 4m 35s
285:	learn: 0.4230483	total: 2m 33s	remaining: 4m 35s
286:	learn: 0.4231604	total: 2m 33s	remaining: 4m 34s
287:	learn: 0.4232143	total: 2m 34s	remaining: 4m 34s
288:	learn: 0.4233309	total: 2m 34s	remaining: 4m 33s
289:	learn: 0.4233623	total: 2m 35s	remaining: 4m 32s
290:	learn: 0.4234834	total: 2m 35s	remaining: 4m 32s
291:	learn: 0.4234341	total: 2m 36s	remaining: 4m 31s
292:	learn: 0.4234520	total: 2m 36s	remaining: 4m 31s
293:	learn: 0.4235462	total: 2m 37s	remaining: 4m 30s
294:	learn: 0.4236719	total: 2m 37s	remaining: 4m 30s
295:	learn: 0.4237436	total: 2m 38s	remaining: 4m 29s
296:	learn: 0.4239051	total: 2m 38s	remaining: 4m 28s
297:	learn: 0.4240083	total: 2m 39s	remaining: 4m 28s
298:	learn: 0.4241653	total: 2m 39s	remaining: 4m 27s
299:	learn: 0.4241743	total: 2m 40s	remaining: 4m 27s
300:	learn: 0.4242371	total: 2m 40s	remaining: 4m 26s
301:	learn: 0.4241743	total: 2m 41s	remaining: 4m 26s
302:	learn: 0.4242775	total: 2m 41s	remaining: 4m 25s
303:	learn: 0.4243134	total: 2m 42s	remaining: 4m 25s
304:	learn: 0.4243493	total: 2m 43s	remaining: 4m 25s
305:	learn: 0.4243627	total: 2m 44s	remaining: 4m 24s
306:	learn: 0.4243941	total: 2m 44s	remaining: 4m 24s
307:	learn: 0.4244480	total: 2m 45s	remaining: 4m 23s
308:	learn: 0.4247127	total: 2m 45s	remaining: 4m 23s
309:	learn: 0.4247979	total: 2m 46s	remaining: 4m 22s
310:	learn: 0.4248831	total: 2m 46s	remaining: 4m 21s
311:	learn: 0.4249549	total: 2m 47s	remaining: 4m 21s
312:	learn: 0.4249370	total: 2m 47s	remaining: 4m 20s
```

```
313:	learn: 0.4250267	total: 2m 48s	remaining: 4m 20s
314:	learn: 0.4251030	total: 2m 48s	remaining: 4m 19s
315:	learn: 0.4252914	total: 2m 49s	remaining: 4m 19s
316:	learn: 0.4253004	total: 2m 49s	remaining: 4m 18s
317:	learn: 0.4253452	total: 2m 50s	remaining: 4m 18s
318:	learn: 0.4254529	total: 2m 50s	remaining: 4m 17s
319:	learn: 0.4255067	total: 2m 51s	remaining: 4m 16s
320:	learn: 0.4255650	total: 2m 51s	remaining: 4m 16s
321:	learn: 0.4256817	total: 2m 52s	remaining: 4m 15s
322:	learn: 0.4257804	total: 2m 52s	remaining: 4m 15s
323:	learn: 0.4258432	total: 2m 53s	remaining: 4m 14s
324:	learn: 0.4261124	total: 2m 53s	remaining: 4m 14s
325:	learn: 0.4261034	total: 2m 54s	remaining: 4m 13s
326:	learn: 0.4263367	total: 2m 54s	remaining: 4m 12s
327:	learn: 0.4263546	total: 2m 55s	remaining: 4m 12s
328:	learn: 0.4264892	total: 2m 55s	remaining: 4m 11s
329:	learn: 0.4265565	total: 2m 56s	remaining: 4m 11s
330:	learn: 0.4266193	total: 2m 56s	remaining: 4m 10s
331:	learn: 0.4267987	total: 2m 57s	remaining: 4m 10s
332:	learn: 0.4267629	total: 2m 57s	remaining: 4m 9s
333:	learn: 0.4268840	total: 2m 58s	remaining: 4m 8s
334:	learn: 0.4270320	total: 2m 58s	remaining: 4m 8s
335:	learn: 0.4270679	total: 2m 59s	remaining: 4m 7s
336:	learn: 0.4271935	total: 2m 59s	remaining: 4m 7s
337:	learn: 0.4273685	total: 3m	remaining: 4m 6s
338:	learn: 0.4273461	total: 3m	remaining: 4m 6s
339:	learn: 0.4275210	total: 3m 1s	remaining: 4m 5s
340:	learn: 0.4276825	total: 3m 2s	remaining: 4m 5s
341:	learn: 0.4277543	total: 3m 2s	remaining: 4m 4s
342:	learn: 0.4278889	total: 3m 3s	remaining: 4m 4s
343:	learn: 0.4280683	total: 3m 3s	remaining: 4m 3s
344:	learn: 0.4281311	total: 3m 4s	remaining: 4m 3s
345:	learn: 0.4282074	total: 3m 4s	remaining: 4m 2s
346:	learn: 0.4282747	total: 3m 5s	remaining: 4m 2s
347:	learn: 0.4282254	total: 3m 5s	remaining: 4m 1s
348:	learn: 0.4282971	total: 3m 6s	remaining: 4m
349:	learn: 0.4283375	total: 3m 6s	remaining: 4m
350:	learn: 0.4282971	total: 3m 7s	remaining: 3m 59s
351:	learn: 0.4283465	total: 3m 7s	remaining: 3m 59s
352:	learn: 0.4285125	total: 3m 8s	remaining: 3m 58s
353:	learn: 0.4285977	total: 3m 8s	remaining: 3m 58s
354:	learn: 0.4287099	total: 3m 9s	remaining: 3m 57s
355:	learn: 0.4287861	total: 3m 9s	remaining: 3m 56s
356:	learn: 0.4288310	total: 3m 10s	remaining: 3m 56s
357:	learn: 0.4291540	total: 3m 10s	remaining: 3m 55s
358:	learn: 0.4291316	total: 3m 11s	remaining: 3m 55s
359:	learn: 0.4291809	total: 3m 11s	remaining: 3m 54s
360:	learn: 0.4291495	total: 3m 12s	remaining: 3m 54s
361:	learn: 0.4292123	total: 3m 13s	remaining: 3m 53s
362:	learn: 0.4293379	total: 3m 13s	remaining: 3m 53s
363:	learn: 0.4293334	total: 3m 14s	remaining: 3m 52s
364:	learn: 0.4294321	total: 3m 14s	remaining: 3m 52s
365:	learn: 0.4295174	total: 3m 15s	remaining: 3m 51s
366:	learn: 0.4295219	total: 3m 16s	remaining: 3m 51s
367:	learn: 0.4297327	total: 3m 16s	remaining: 3m 50s
368:	learn: 0.4298897	total: 3m 17s	remaining: 3m 50s
369:	learn: 0.4299795	total: 3m 18s	remaining: 3m 50s
370:	learn: 0.4301679	total: 3m 18s	remaining: 3m 49s
371:	learn: 0.4302082	total: 3m 19s	remaining: 3m 49s
372:	learn: 0.4304909	total: 3m 19s	remaining: 3m 48s
373:	learn: 0.4305357	total: 3m 20s	remaining: 3m 48s
374:	learn: 0.4305851	total: 3m 20s	remaining: 3m 47s
375:	learn: 0.4306524	total: 3m 21s	remaining: 3m 46s
376:	learn: 0.4306883	total: 3m 21s	remaining: 3m 46s
377:	learn: 0.4307601	total: 3m 22s	remaining: 3m 45s
```

```
378:	learn: 0.4308363	total: 3m 22s	remaining: 3m 45s
379:	learn: 0.4309081	total: 3m 23s	remaining: 3m 44s
380:	learn: 0.4310696	total: 3m 24s	remaining: 3m 44s
381:	learn: 0.4312535	total: 3m 24s	remaining: 3m 43s
382:	learn: 0.4312670	total: 3m 25s	remaining: 3m 43s
383:	learn: 0.4313208	total: 3m 25s	remaining: 3m 42s
384:	learn: 0.4313567	total: 3m 26s	remaining: 3m 42s
385:	learn: 0.4314778	total: 3m 26s	remaining: 3m 41s
386:	learn: 0.4314285	total: 3m 27s	remaining: 3m 41s
387:	learn: 0.4315945	total: 3m 27s	remaining: 3m 40s
388:	learn: 0.4316349	total: 3m 28s	remaining: 3m 40s
389:	learn: 0.4316797	total: 3m 28s	remaining: 3m 39s
390:	learn: 0.4318008	total: 3m 29s	remaining: 3m 38s
391:	learn: 0.4319534	total: 3m 29s	remaining: 3m 38s
392:	learn: 0.4321867	total: 3m 30s	remaining: 3m 37s
393:	learn: 0.4322988	total: 3m 30s	remaining: 3m 37s
394:	learn: 0.4323437	total: 3m 31s	remaining: 3m 36s
395:	learn: 0.4322898	total: 3m 31s	remaining: 3m 36s
396:	learn: 0.4324917	total: 3m 32s	remaining: 3m 35s
397:	learn: 0.4325276	total: 3m 33s	remaining: 3m 35s
398:	learn: 0.4325142	total: 3m 33s	remaining: 3m 34s
399:	learn: 0.4325635	total: 3m 34s	remaining: 3m 34s
400:	learn: 0.4326712	total: 3m 34s	remaining: 3m 33s
401:	learn: 0.4328955	total: 3m 35s	remaining: 3m 33s
402:	learn: 0.4329179	total: 3m 35s	remaining: 3m 32s
403:	learn: 0.4329762	total: 3m 36s	remaining: 3m 32s
404:	learn: 0.4330615	total: 3m 37s	remaining: 3m 31s
405:	learn: 0.4331557	total: 3m 37s	remaining: 3m 31s
406:	learn: 0.4332454	total: 3m 38s	remaining: 3m 30s
407:	learn: 0.4333396	total: 3m 38s	remaining: 3m 30s
408:	learn: 0.4334563	total: 3m 39s	remaining: 3m 29s
409:	learn: 0.4335191	total: 3m 39s	remaining: 3m 28s
410:	learn: 0.4337344	total: 3m 40s	remaining: 3m 28s
411:	learn: 0.4337209	total: 3m 40s	remaining: 3m 27s
412:	learn: 0.4338017	total: 3m 41s	remaining: 3m 27s
413:	learn: 0.4338735	total: 3m 41s	remaining: 3m 26s
414:	learn: 0.4338780	total: 3m 42s	remaining: 3m 26s
415:	learn: 0.4340170	total: 3m 42s	remaining: 3m 25s
416:	learn: 0.4340888	total: 3m 43s	remaining: 3m 25s
417:	learn: 0.4343535	total: 3m 43s	remaining: 3m 24s
418:	learn: 0.4344028	total: 3m 44s	remaining: 3m 23s
419:	learn: 0.4345060	total: 3m 44s	remaining: 3m 23s
420:	learn: 0.4345150	total: 3m 45s	remaining: 3m 22s
421:	learn: 0.4346541	total: 3m 45s	remaining: 3m 22s
422:	learn: 0.4348021	total: 3m 46s	remaining: 3m 21s
423:	learn: 0.4348604	total: 3m 46s	remaining: 3m 21s
424:	learn: 0.4349053	total: 3m 47s	remaining: 3m 20s
425:	learn: 0.4349502	total: 3m 48s	remaining: 3m 20s
426:	learn: 0.4350937	total: 3m 48s	remaining: 3m 19s
427:	learn: 0.4350758	total: 3m 49s	remaining: 3m 19s
428:	learn: 0.4350937	total: 3m 50s	remaining: 3m 18s
429:	learn: 0.4351610	total: 3m 50s	remaining: 3m 18s
430:	learn: 0.4354033	total: 3m 51s	remaining: 3m 17s
431:	learn: 0.4354885	total: 3m 51s	remaining: 3m 17s
432:	learn: 0.4354302	total: 3m 52s	remaining: 3m 16s
433:	learn: 0.4355558	total: 3m 52s	remaining: 3m 16s
434:	learn: 0.4357442	total: 3m 53s	remaining: 3m 15s
435:	learn: 0.4358474	total: 3m 53s	remaining: 3m 15s
436:	learn: 0.4358609	total: 3m 54s	remaining: 3m 14s
437:	learn: 0.4359102	total: 3m 54s	remaining: 3m 13s
438:	learn: 0.4360089	total: 3m 55s	remaining: 3m 13s
439:	learn: 0.4359730	total: 3m 55s	remaining: 3m 12s
440:	learn: 0.4360493	total: 3m 56s	remaining: 3m 12s
441:	learn: 0.4361614	total: 3m 56s	remaining: 3m 11s
442:	learn: 0.4362781	total: 3m 57s	remaining: 3m 11s
```

```
443:	learn: 0.4362915	total: 3m 57s	remaining: 3m 10s
444:	learn: 0.4364261	total: 3m 58s	remaining: 3m 10s
445:	learn: 0.4363768	total: 3m 58s	remaining: 3m 9s
446:	learn: 0.4365517	total: 3m 59s	remaining: 3m 8s
447:	learn: 0.4366415	total: 3m 59s	remaining: 3m 8s
448:	learn: 0.4368299	total: 4m	remaining: 3m 7s
449:	learn: 0.4368613	total: 4m	remaining: 3m 7s
450:	learn: 0.4370048	total: 4m 1s	remaining: 3m 6s
451:	learn: 0.4371080	total: 4m 1s	remaining: 3m 6s
452:	learn: 0.4372426	total: 4m 2s	remaining: 3m 5s
453:	learn: 0.4374849	total: 4m 2s	remaining: 3m 5s
454:	learn: 0.4376912	total: 4m 3s	remaining: 3m 4s
455:	learn: 0.4376912	total: 4m 3s	remaining: 3m 3s
456:	learn: 0.4376867	total: 4m 4s	remaining: 3m 3s
457:	learn: 0.4378079	total: 4m 4s	remaining: 3m 2s
458:	learn: 0.4378438	total: 4m 5s	remaining: 3m 2s
459:	learn: 0.4379245	total: 4m 5s	remaining: 3m 1s
460:	learn: 0.4379335	total: 4m 6s	remaining: 3m 1s
461:	learn: 0.4380995	total: 4m 6s	remaining: 3m
462:	learn: 0.4381040	total: 4m 7s	remaining: 3m
463:	learn: 0.4381309	total: 4m 8s	remaining: 2m 59s
464:	learn: 0.4380770	total: 4m 8s	remaining: 2m 59s
465:	learn: 0.4381937	total: 4m 9s	remaining: 2m 58s
466:	learn: 0.4382430	total: 4m 9s	remaining: 2m 58s
467:	learn: 0.4382969	total: 4m 10s	remaining: 2m 57s
468:	learn: 0.4384898	total: 4m 10s	remaining: 2m 57s
469:	learn: 0.4385481	total: 4m 11s	remaining: 2m 56s
470:	learn: 0.4385750	total: 4m 11s	remaining: 2m 55s
471:	learn: 0.4386199	total: 4m 12s	remaining: 2m 55s
472:	learn: 0.4386961	total: 4m 12s	remaining: 2m 54s
473:	learn: 0.4387634	total: 4m 13s	remaining: 2m 54s
474:	learn: 0.4388083	total: 4m 13s	remaining: 2m 53s
475:	learn: 0.4388890	total: 4m 14s	remaining: 2m 53s
476:	learn: 0.4389204	total: 4m 14s	remaining: 2m 52s
477:	learn: 0.4389384	total: 4m 15s	remaining: 2m 52s
478:	learn: 0.4392479	total: 4m 15s	remaining: 2m 51s
479:	learn: 0.4393152	total: 4m 16s	remaining: 2m 50s
480:	learn: 0.4392390	total: 4m 17s	remaining: 2m 50s
481:	learn: 0.4393780	total: 4m 17s	remaining: 2m 50s
482:	learn: 0.4395261	total: 4m 18s	remaining: 2m 49s
483:	learn: 0.4395440	total: 4m 18s	remaining: 2m 48s
484:	learn: 0.4396293	total: 4m 19s	remaining: 2m 48s
485:	learn: 0.4396517	total: 4m 20s	remaining: 2m 48s
486:	learn: 0.4397459	total: 4m 20s	remaining: 2m 47s
487:	learn: 0.4398446	total: 4m 21s	remaining: 2m 47s
488:	learn: 0.4399657	total: 4m 22s	remaining: 2m 46s
489:	learn: 0.4400510	total: 4m 22s	remaining: 2m 46s
490:	learn: 0.4402798	total: 4m 23s	remaining: 2m 45s
491:	learn: 0.4403560	total: 4m 23s	remaining: 2m 45s
492:	learn: 0.4404233	total: 4m 24s	remaining: 2m 44s
493:	learn: 0.4405758	total: 4m 24s	remaining: 2m 44s
494:	learn: 0.4406028	total: 4m 25s	remaining: 2m 43s
495:	learn: 0.4404996	total: 4m 26s	remaining: 2m 43s
496:	learn: 0.4406476	total: 4m 26s	remaining: 2m 42s
497:	learn: 0.4407957	total: 4m 27s	remaining: 2m 41s
498:	learn: 0.4408944	total: 4m 27s	remaining: 2m 41s
499:	learn: 0.4409527	total: 4m 28s	remaining: 2m 40s
500:	learn: 0.4409976	total: 4m 28s	remaining: 2m 40s
501:	learn: 0.4411142	total: 4m 29s	remaining: 2m 39s
502:	learn: 0.4411680	total: 4m 29s	remaining: 2m 39s
503:	learn: 0.4412802	total: 4m 30s	remaining: 2m 38s
504:	learn: 0.4413250	total: 4m 30s	remaining: 2m 38s
505:	learn: 0.4413699	total: 4m 31s	remaining: 2m 37s
506:	learn: 0.4415449	total: 4m 31s	remaining: 2m 37s
507:	learn: 0.4415942	total: 4m 32s	remaining: 2m 36s
```

```
508:	learn: 0.4416750	total: 4m 32s	remaining: 2m 35s
509:	learn: 0.4417602	total: 4m 33s	remaining: 2m 35s
510:	learn: 0.4418320	total: 4m 33s	remaining: 2m 34s
511:	learn: 0.4419038	total: 4m 34s	remaining: 2m 34s
512:	learn: 0.4420294	total: 4m 34s	remaining: 2m 33s
513:	learn: 0.4421326	total: 4m 35s	remaining: 2m 33s
514:	learn: 0.4421999	total: 4m 35s	remaining: 2m 32s
515:	learn: 0.4421191	total: 4m 36s	remaining: 2m 32s
516:	learn: 0.4421819	total: 4m 36s	remaining: 2m 31s
517:	learn: 0.4424197	total: 4m 37s	remaining: 2m 30s
518:	learn: 0.4425004	total: 4m 37s	remaining: 2m 30s
519:	learn: 0.4425318	total: 4m 38s	remaining: 2m 29s
520:	learn: 0.4425139	total: 4m 38s	remaining: 2m 29s
521:	learn: 0.4426216	total: 4m 39s	remaining: 2m 28s
522:	learn: 0.4426305	total: 4m 39s	remaining: 2m 28s
523:	learn: 0.4426530	total: 4m 40s	remaining: 2m 27s
524:	learn: 0.4428189	total: 4m 41s	remaining: 2m 27s
525:	learn: 0.4430163	total: 4m 41s	remaining: 2m 26s
526:	learn: 0.4430343	total: 4m 42s	remaining: 2m 26s
527:	learn: 0.4430253	total: 4m 42s	remaining: 2m 25s
528:	learn: 0.4432721	total: 4m 43s	remaining: 2m 25s
529:	learn: 0.4433932	total: 4m 43s	remaining: 2m 24s
530:	learn: 0.4435995	total: 4m 44s	remaining: 2m 24s
531:	learn: 0.4436085	total: 4m 44s	remaining: 2m 23s
532:	learn: 0.4436624	total: 4m 45s	remaining: 2m 22s
533:	learn: 0.4437117	total: 4m 45s	remaining: 2m 22s
534:	learn: 0.4436893	total: 4m 46s	remaining: 2m 21s
535:	learn: 0.4438014	total: 4m 46s	remaining: 2m 21s
536:	learn: 0.4438687	total: 4m 47s	remaining: 2m 20s
537:	learn: 0.4438911	total: 4m 47s	remaining: 2m 20s
538:	learn: 0.4438777	total: 4m 48s	remaining: 2m 19s
539:	learn: 0.4438777	total: 4m 48s	remaining: 2m 19s
540:	learn: 0.4438867	total: 4m 49s	remaining: 2m 18s
541:	learn: 0.4438732	total: 4m 49s	remaining: 2m 17s
542:	learn: 0.4440212	total: 4m 50s	remaining: 2m 17s
543:	learn: 0.4440975	total: 4m 50s	remaining: 2m 16s
544:	learn: 0.4440930	total: 4m 51s	remaining: 2m 16s
545:	learn: 0.4442276	total: 4m 52s	remaining: 2m 15s
546:	learn: 0.4442814	total: 4m 52s	remaining: 2m 15s
547:	learn: 0.4443757	total: 4m 53s	remaining: 2m 14s
548:	learn: 0.4444609	total: 4m 53s	remaining: 2m 14s
549:	learn: 0.4445551	total: 4m 54s	remaining: 2m 13s
550:	learn: 0.4446717	total: 4m 54s	remaining: 2m 13s
551:	learn: 0.4447031	total: 4m 55s	remaining: 2m 12s
552:	learn: 0.4446717	total: 4m 56s	remaining: 2m 12s
553:	learn: 0.4449903	total: 4m 56s	remaining: 2m 11s
554:	learn: 0.4450800	total: 4m 57s	remaining: 2m 11s
555:	learn: 0.4450710	total: 4m 57s	remaining: 2m 10s
556:	learn: 0.4451607	total: 4m 58s	remaining: 2m 10s
557:	learn: 0.4452908	total: 4m 58s	remaining: 2m 9s
558:	learn: 0.4454030	total: 4m 59s	remaining: 2m 9s
559:	learn: 0.4454568	total: 4m 59s	remaining: 2m 8s
560:	learn: 0.4456363	total: 5m	remaining: 2m 7s
561:	learn: 0.4455959	total: 5m	remaining: 2m 7s
562:	learn: 0.4457574	total: 5m 1s	remaining: 2m 6s
563:	learn: 0.4457753	total: 5m 1s	remaining: 2m 6s
564:	learn: 0.4458785	total: 5m 2s	remaining: 2m 5s
565:	learn: 0.4458875	total: 5m 2s	remaining: 2m 5s
566:	learn: 0.4459772	total: 5m 3s	remaining: 2m 4s
567:	learn: 0.4460714	total: 5m 3s	remaining: 2m 4s
568:	learn: 0.4461522	total: 5m 4s	remaining: 2m 3s
569:	learn: 0.4462464	total: 5m 4s	remaining: 2m 3s
570:	learn: 0.4463855	total: 5m 5s	remaining: 2m 2s
571:	learn: 0.4463900	total: 5m 5s	remaining: 2m 1s
572:	learn: 0.4464707	total: 5m 6s	remaining: 2m 1s
```

```
573:	learn: 0.4464976	total: 5m 7s	remaining: 2m
574:	learn: 0.4466277	total: 5m 7s	remaining: 2m
575:	learn: 0.4466277	total: 5m 8s	remaining: 1m 59s
576:	learn: 0.4466816	total: 5m 8s	remaining: 1m 59s
577:	learn: 0.4466995	total: 5m 9s	remaining: 1m 58s
578:	learn: 0.4467085	total: 5m 9s	remaining: 1m 58s
579:	learn: 0.4467040	total: 5m 10s	remaining: 1m 57s
580:	learn: 0.4468027	total: 5m 10s	remaining: 1m 57s
581:	learn: 0.4470270	total: 5m 11s	remaining: 1m 56s
582:	learn: 0.4470539	total: 5m 11s	remaining: 1m 56s
583:	learn: 0.4470539	total: 5m 12s	remaining: 1m 55s
584:	learn: 0.4472737	total: 5m 12s	remaining: 1m 54s
585:	learn: 0.4474173	total: 5m 13s	remaining: 1m 54s
586:	learn: 0.4475160	total: 5m 14s	remaining: 1m 53s
587:	learn: 0.4475519	total: 5m 14s	remaining: 1m 53s
588:	learn: 0.4474980	total: 5m 15s	remaining: 1m 52s
589:	learn: 0.4476551	total: 5m 15s	remaining: 1m 52s
590:	learn: 0.4476910	total: 5m 16s	remaining: 1m 51s
591:	learn: 0.4479108	total: 5m 16s	remaining: 1m 51s
592:	learn: 0.4478435	total: 5m 17s	remaining: 1m 50s
593:	learn: 0.4478255	total: 5m 18s	remaining: 1m 50s
594:	learn: 0.4480184	total: 5m 18s	remaining: 1m 49s
595:	learn: 0.4479556	total: 5m 19s	remaining: 1m 49s
596:	learn: 0.4481396	total: 5m 19s	remaining: 1m 48s
597:	learn: 0.4482024	total: 5m 20s	remaining: 1m 48s
598:	learn: 0.4482248	total: 5m 20s	remaining: 1m 47s
599:	learn: 0.4482383	total: 5m 21s	remaining: 1m 47s
600:	learn: 0.4482831	total: 5m 21s	remaining: 1m 46s
601:	learn: 0.4484222	total: 5m 22s	remaining: 1m 46s
602:	learn: 0.4485658	total: 5m 22s	remaining: 1m 45s
603:	learn: 0.4486375	total: 5m 23s	remaining: 1m 44s
604:	learn: 0.4486286	total: 5m 23s	remaining: 1m 44s
605:	learn: 0.4487183	total: 5m 24s	remaining: 1m 43s
606:	learn: 0.4490458	total: 5m 25s	remaining: 1m 43s
607:	learn: 0.4489875	total: 5m 25s	remaining: 1m 42s
608:	learn: 0.4490503	total: 5m 26s	remaining: 1m 42s
609:	learn: 0.4490906	total: 5m 26s	remaining: 1m 41s
610:	learn: 0.4492073	total: 5m 27s	remaining: 1m 41s
611:	learn: 0.4492970	total: 5m 28s	remaining: 1m 40s
612:	learn: 0.4493867	total: 5m 28s	remaining: 1m 40s
613:	learn: 0.4495707	total: 5m 29s	remaining: 1m 39s
614:	learn: 0.4495976	total: 5m 29s	remaining: 1m 39s
615:	learn: 0.4495707	total: 5m 30s	remaining: 1m 38s
616:	learn: 0.4495841	total: 5m 30s	remaining: 1m 38s
617:	learn: 0.4495796	total: 5m 31s	remaining: 1m 37s
618:	learn: 0.4497546	total: 5m 32s	remaining: 1m 37s
619:	learn: 0.4498443	total: 5m 32s	remaining: 1m 36s
620:	learn: 0.4498309	total: 5m 33s	remaining: 1m 35s
621:	learn: 0.4498668	total: 5m 33s	remaining: 1m 35s
622:	learn: 0.4500283	total: 5m 34s	remaining: 1m 34s
623:	learn: 0.4500597	total: 5m 34s	remaining: 1m 34s
624:	learn: 0.4501808	total: 5m 35s	remaining: 1m 33s
625:	learn: 0.4503333	total: 5m 35s	remaining: 1m 33s
626:	learn: 0.4504275	total: 5m 36s	remaining: 1m 32s
627:	learn: 0.4505128	total: 5m 36s	remaining: 1m 32s
628:	learn: 0.4505173	total: 5m 37s	remaining: 1m 31s
629:	learn: 0.4505890	total: 5m 37s	remaining: 1m 31s
630:	learn: 0.4507505	total: 5m 38s	remaining: 1m 30s
631:	learn: 0.4507954	total: 5m 38s	remaining: 1m 30s
632:	learn: 0.4508178	total: 5m 39s	remaining: 1m 29s
633:	learn: 0.4508717	total: 5m 39s	remaining: 1m 28s
634:	learn: 0.4508896	total: 5m 40s	remaining: 1m 28s
635:	learn: 0.4510018	total: 5m 40s	remaining: 1m 27s
636:	learn: 0.4511453	total: 5m 41s	remaining: 1m 27s
637:	learn: 0.4511767	total: 5m 41s	remaining: 1m 26s
```

```
638:	learn: 0.4512216	total: 5m 42s	remaining: 1m 26s
639:	learn: 0.4513786	total: 5m 42s	remaining: 1m 25s
640:	learn: 0.4515805	total: 5m 43s	remaining: 1m 25s
641:	learn: 0.4516523	total: 5m 43s	remaining: 1m 24s
642:	learn: 0.4516612	total: 5m 44s	remaining: 1m 24s
643:	learn: 0.4516209	total: 5m 44s	remaining: 1m 23s
644:	learn: 0.4517779	total: 5m 45s	remaining: 1m 23s
645:	learn: 0.4518452	total: 5m 45s	remaining: 1m 22s
646:	learn: 0.4518586	total: 5m 46s	remaining: 1m 21s
647:	learn: 0.4520471	total: 5m 47s	remaining: 1m 21s
648:	learn: 0.4519887	total: 5m 47s	remaining: 1m 20s
649:	learn: 0.4520874	total: 5m 48s	remaining: 1m 20s
650:	learn: 0.4521188	total: 5m 48s	remaining: 1m 19s
651:	learn: 0.4522669	total: 5m 49s	remaining: 1m 19s
652:	learn: 0.4522938	total: 5m 49s	remaining: 1m 18s
653:	learn: 0.4523701	total: 5m 50s	remaining: 1m 18s
654:	learn: 0.4524688	total: 5m 50s	remaining: 1m 17s
655:	learn: 0.4525540	total: 5m 51s	remaining: 1m 17s
656:	learn: 0.4526213	total: 5m 51s	remaining: 1m 16s
657:	learn: 0.4527424	total: 5m 52s	remaining: 1m 16s
658:	learn: 0.4528007	total: 5m 52s	remaining: 1m 15s
659:	learn: 0.4529398	total: 5m 53s	remaining: 1m 14s
660:	learn: 0.4529667	total: 5m 53s	remaining: 1m 14s
661:	learn: 0.4530161	total: 5m 54s	remaining: 1m 13s
662:	learn: 0.4531507	total: 5m 55s	remaining: 1m 13s
663:	learn: 0.4531641	total: 5m 55s	remaining: 1m 12s
664:	learn: 0.4532359	total: 5m 56s	remaining: 1m 12s
665:	learn: 0.4533032	total: 5m 56s	remaining: 1m 11s
666:	learn: 0.4533705	total: 5m 57s	remaining: 1m 11s
667:	learn: 0.4534781	total: 5m 57s	remaining: 1m 10s
668:	learn: 0.4535230	total: 5m 58s	remaining: 1m 10s
669:	learn: 0.4535993	total: 5m 58s	remaining: 1m 9s
670:	learn: 0.4536486	total: 5m 59s	remaining: 1m 9s
671:	learn: 0.4536845	total: 5m 59s	remaining: 1m 8s
672:	learn: 0.4538012	total: 6m	remaining: 1m 8s
673:	learn: 0.4538729	total: 6m 1s	remaining: 1m 7s
674:	learn: 0.4539043	total: 6m 1s	remaining: 1m 7s
675:	learn: 0.4539223	total: 6m 2s	remaining: 1m 6s
676:	learn: 0.4539806	total: 6m 3s	remaining: 1m 5s
677:	learn: 0.4539671	total: 6m 3s	remaining: 1m 5s
678:	learn: 0.4540479	total: 6m 4s	remaining: 1m 4s
679:	learn: 0.4541242	total: 6m 4s	remaining: 1m 4s
680:	learn: 0.4542273	total: 6m 5s	remaining: 1m 3s
681:	learn: 0.4541959	total: 6m 5s	remaining: 1m 3s
682:	learn: 0.4542901	total: 6m 6s	remaining: 1m 2s
683:	learn: 0.4545369	total: 6m 6s	remaining: 1m 2s
684:	learn: 0.4546356	total: 6m 7s	remaining: 1m 1s
685:	learn: 0.4546221	total: 6m 7s	remaining: 1m 1s
686:	learn: 0.4546446	total: 6m 8s	remaining: 1m
687:	learn: 0.4547657	total: 6m 8s	remaining: 1m
688:	learn: 0.4549990	total: 6m 9s	remaining: 59.5s
689:	learn: 0.4549765	total: 6m 9s	remaining: 58.9s
690:	learn: 0.4550932	total: 6m 10s	remaining: 58.4s
691:	learn: 0.4552053	total: 6m 10s	remaining: 57.9s
692:	learn: 0.4551784	total: 6m 11s	remaining: 57.3s
693:	learn: 0.4552278	total: 6m 11s	remaining: 56.8s
694:	learn: 0.4552592	total: 6m 12s	remaining: 56.2s
695:	learn: 0.4553265	total: 6m 12s	remaining: 55.7s
696:	learn: 0.4553265	total: 6m 13s	remaining: 55.2s
697:	learn: 0.4554566	total: 6m 13s	remaining: 54.6s
698:	learn: 0.4556360	total: 6m 14s	remaining: 54.1s
699:	learn: 0.4556540	total: 6m 14s	remaining: 53.5s
700:	learn: 0.4557078	total: 6m 15s	remaining: 53s
701:	learn: 0.4557078	total: 6m 15s	remaining: 52.5s
702:	learn: 0.4557033	total: 6m 16s	remaining: 51.9s
```

```
703:	learn: 0.4557571	total: 6m 16s	remaining: 51.4s
704:	learn: 0.4558469	total: 6m 17s	remaining: 50.9s
705:	learn: 0.4559904	total: 6m 18s	remaining: 50.3s
706:	learn: 0.4560218	total: 6m 18s	remaining: 49.8s
707:	learn: 0.4561071	total: 6m 19s	remaining: 49.3s
708:	learn: 0.4562237	total: 6m 20s	remaining: 48.8s
709:	learn: 0.4562416	total: 6m 20s	remaining: 48.3s
710:	learn: 0.4562686	total: 6m 21s	remaining: 47.7s
711:	learn: 0.4564390	total: 6m 21s	remaining: 47.2s
712:	learn: 0.4565781	total: 6m 22s	remaining: 46.7s
713:	learn: 0.4566499	total: 6m 22s	remaining: 46.1s
714:	learn: 0.4567486	total: 6m 23s	remaining: 45.6s
715:	learn: 0.4567710	total: 6m 24s	remaining: 45.1s
716:	learn: 0.4567351	total: 6m 24s	remaining: 44.5s
717:	learn: 0.4567890	total: 6m 25s	remaining: 44s
718:	learn: 0.4568383	total: 6m 25s	remaining: 43.4s
719:	learn: 0.4569011	total: 6m 26s	remaining: 42.9s
720:	learn: 0.4570178	total: 6m 26s	remaining: 42.4s
721:	learn: 0.4570761	total: 6m 27s	remaining: 41.8s
722:	learn: 0.4570626	total: 6m 27s	remaining: 41.3s
723:	learn: 0.4572196	total: 6m 28s	remaining: 40.7s
724:	learn: 0.4572555	total: 6m 28s	remaining: 40.2s
725:	learn: 0.4575112	total: 6m 29s	remaining: 39.7s
726:	learn: 0.4574125	total: 6m 29s	remaining: 39.1s
727:	learn: 0.4575696	total: 6m 30s	remaining: 38.6s
728:	learn: 0.4576010	total: 6m 30s	remaining: 38.1s
729:	learn: 0.4576683	total: 6m 31s	remaining: 37.5s
730:	learn: 0.4578028	total: 6m 31s	remaining: 37s
731:	learn: 0.4579240	total: 6m 32s	remaining: 36.5s
732:	learn: 0.4583681	total: 6m 32s	remaining: 35.9s
733:	learn: 0.4583726	total: 6m 33s	remaining: 35.4s
734:	learn: 0.4584309	total: 6m 34s	remaining: 34.9s
735:	learn: 0.4585296	total: 6m 35s	remaining: 34.3s
736:	learn: 0.4585475	total: 6m 35s	remaining: 33.8s
737:	learn: 0.4585924	total: 6m 36s	remaining: 33.3s
738:	learn: 0.4585475	total: 6m 36s	remaining: 32.7s
739:	learn: 0.4586238	total: 6m 37s	remaining: 32.2s
740:	learn: 0.4586776	total: 6m 37s	remaining: 31.7s
741:	learn: 0.4588706	total: 6m 38s	remaining: 31.1s
742:	learn: 0.4590141	total: 6m 38s	remaining: 30.6s
743:	learn: 0.4590231	total: 6m 39s	remaining: 30s
744:	learn: 0.4592205	total: 6m 39s	remaining: 29.5s
745:	learn: 0.4592519	total: 6m 40s	remaining: 29s
746:	learn: 0.4593461	total: 6m 40s	remaining: 28.4s
747:	learn: 0.4593237	total: 6m 41s	remaining: 27.9s
748:	learn: 0.4593192	total: 6m 41s	remaining: 27.4s
749:	learn: 0.4594268	total: 6m 42s	remaining: 26.8s
750:	learn: 0.4594448	total: 6m 42s	remaining: 26.3s
751:	learn: 0.4594313	total: 6m 43s	remaining: 25.8s
752:	learn: 0.4595435	total: 6m 44s	remaining: 25.2s
753:	learn: 0.4596242	total: 6m 44s	remaining: 24.7s
754:	learn: 0.4597723	total: 6m 45s	remaining: 24.1s
755:	learn: 0.4598441	total: 6m 45s	remaining: 23.6s
756:	learn: 0.4599069	total: 6m 46s	remaining: 23.1s
757:	learn: 0.4599697	total: 6m 46s	remaining: 22.5s
758:	learn: 0.4600594	total: 6m 47s	remaining: 22s
759:	learn: 0.4602837	total: 6m 47s	remaining: 21.5s
760:	learn: 0.4604273	total: 6m 48s	remaining: 20.9s
761:	learn: 0.4604183	total: 6m 48s	remaining: 20.4s
762:	learn: 0.4605439	total: 6m 49s	remaining: 19.8s
763:	learn: 0.4606157	total: 6m 49s	remaining: 19.3s
764:	learn: 0.4607144	total: 6m 50s	remaining: 18.8s
765:	learn: 0.4607503	total: 6m 50s	remaining: 18.2s
766:	learn: 0.4608669	total: 6m 51s	remaining: 17.7s
767:	learn: 0.4610150	total: 6m 52s	remaining: 17.2s
```

```
768:	learn: 0.4610284	total: 6m 52s	remaining: 16.6s
769:	learn: 0.4608983	total: 6m 53s	remaining: 16.1s
770:	learn: 0.4609566	total: 6m 53s	remaining: 15.6s
771:	learn: 0.4611047	total: 6m 54s	remaining: 15s
772:	learn: 0.4610867	total: 6m 54s	remaining: 14.5s
773:	learn: 0.4612079	total: 6m 55s	remaining: 13.9s
774:	learn: 0.4613963	total: 6m 55s	remaining: 13.4s
775:	learn: 0.4614591	total: 6m 56s	remaining: 12.9s
776:	learn: 0.4616026	total: 6m 56s	remaining: 12.3s
777:	learn: 0.4615937	total: 6m 57s	remaining: 11.8s
778:	learn: 0.4616744	total: 6m 57s	remaining: 11.3s
779:	learn: 0.4617686	total: 6m 58s	remaining: 10.7s
780:	learn: 0.4619032	total: 6m 58s	remaining: 10.2s
781:	learn: 0.4620288	total: 6m 59s	remaining: 9.65s
782:	learn: 0.4620782	total: 6m 59s	remaining: 9.11s
783:	learn: 0.4622442	total: 7m	remaining: 8.58s
784:	learn: 0.4623384	total: 7m	remaining: 8.04s
785:	learn: 0.4623429	total: 7m 1s	remaining: 7.5s
786:	learn: 0.4624281	total: 7m 1s	remaining: 6.97s
787:	learn: 0.4625178	total: 7m 2s	remaining: 6.43s
788:	learn: 0.4625178	total: 7m 2s	remaining: 5.89s
789:	learn: 0.4625717	total: 7m 3s	remaining: 5.36s
790:	learn: 0.4627107	total: 7m 3s	remaining: 4.82s
791:	learn: 0.4627242	total: 7m 4s	remaining: 4.29s
792:	learn: 0.4627197	total: 7m 4s	remaining: 3.75s
793:	learn: 0.4628588	total: 7m 5s	remaining: 3.21s
794:	learn: 0.4629171	total: 7m 6s	remaining: 2.68s
795:	learn: 0.4629126	total: 7m 6s	remaining: 2.14s
796:	learn: 0.4630113	total: 7m 7s	remaining: 1.61s
797:	learn: 0.4630966	total: 7m 8s	remaining: 1.07s
798:	learn: 0.4631908	total: 7m 8s	remaining: 537ms
799:	learn: 0.4631369	total: 7m 9s	remaining: 0us
```

Out[43]: 0.46313692767354847

In [44]:
```
y_preds_cb=cb_clf.predict(X_test)
cb_clf.score(X_test,y_test)
```

Out[44]: 0.40133149101871624

As we can see, catboost seems to have done well at classifying the results better than the other algorithms. We shall use this for our test dataframe.

Testing phase

In [45]:
```
df_test=pd.read_csv('../input/av-healthcare-analytics-ii/healthcare/test_data.cs
df_test.head()

id_df=df_test['patientid']
```

In [46]:
```
df_test['Hospital_type_code_cat']=le.fit_transform(df_test['Hospital_type_code']
df_reg_codes=pd.get_dummies(df_test['Hospital_region_code'])
df_test=pd.merge(df_test,df_reg_codes,on=df_test.index)
df_test.drop('key_0',axis=1,inplace=True)
df_dept=pd.get_dummies(df_test['Department'])
df_test=pd.merge(df_test,df_dept,on=df_test.index)
df_test.drop('key_0',axis=1,inplace=True)

df_test['Ward_Type_cat']=le.fit_transform(df_test['Ward_Type'])
df_test['Ward_Facility_Code_cat']=le.fit_transform(df_test['Ward_Facility_Code']
```

```
df_adm=pd.get_dummies(df_test['Type of Admission'])
df_test=pd.merge(df_test,df_adm,on=df_test.index)
df_test.drop('key_0',axis=1,inplace=True)
df_sev_ill=pd.get_dummies(df_test['Severity of Illness'])
df_test=pd.merge(df_test,df_sev_ill,on=df_test.index)
df_test.drop('key_0',axis=1,inplace=True)


df_test['Age_cat']=le.fit_transform(df_test['Age'])
```

In [47]:
```
unn_cols=['Hospital_type_code','Hospital_region_code','Department','Ward_Type',
          'Ward_Facility_Code','Type of Admission','Severity of Illness','Age',
          'Admission_Deposit','Bed Grade','City_Code_Patient']

for cols in unn_cols:
    df_test.drop(cols,axis=1,inplace=True)
```

In [48]:
```
df_test=df_test[['Hospital_code', 'City_Code_Hospital',
       'Available Extra Rooms in Hospital', 'patientid',
       'Visitors with Patient', 'Hospital_type_code_cat', 'X', 'Y', 'Z',
       'TB & Chest disease', 'anesthesia', 'gynecology', 'radiotherapy',
       'surgery', 'Ward_Type_cat', 'Ward_Facility_Code_cat', 'Emergency',
       'Trauma', 'Urgent', 'Extreme', 'Minor', 'Moderate', 'Age_cat']]
```

In [49]:
```
y_preds=cb_clf.predict(df_test)
```

We now need to convert the numerical stay data into categorical data.

In [50]:
```
df['Stay_cat']=le.fit_transform(df['Stay'])
y_final=le.inverse_transform(y_preds)
```

```
/opt/conda/lib/python3.7/site-packages/sklearn/utils/validation.py:72: DataConve
rsionWarning:

A column-vector y was passed when a 1d array was expected. Please change the sha
pe of y to (n_samples, ), for example using ravel().
```

In [51]:
```
sample_df=pd.read_csv('../input/av-healthcare-analytics-ii/healthcare/sample_sub
df_final=pd.DataFrame(sample_df.iloc[:,0],columns=['case_id'])
df_final['Stay']= y_final
```

In [52]:
```
final_sub=df_final.to_csv('Catboost predictions',index=False)
```

Conclusion

It was seen that from the available data, about 42 % of the cases could be correctly classified. While this is not a very high value, it could definitely mean a situation of life and death maybe avoided by allowing a patient to reach a hospital which definitely has free beds available.

In [ ]: